

AD-A233 602

ENTATION PAGE

Form Approved
OMB No 0704 0188

1a RE unclassified			1b RESTRICTIVE MARKINGS DTIC FILE COPY	
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			5 MONITORING ORGANIZATION REPORT NUMBER(S)	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			7a NAME OF MONITORING ORGANIZATION Cognitive Science Program Office of Naval Research (Code 1142PT)	
6a NAME OF PERFORMING ORGANIZATION The Regents of the University of California		6b OFFICE SYMBOL (if applicable)	7b ADDRESS (City, State, and ZIP Code) 800 North Quincy Street Arlington, VA 22217-5000	
6c ADDRESS (City, State, and ZIP Code) University of California, Los Angeles Office of Contracts and Grants Administration Los Angeles, California 90024		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N00014-86-K-0395		
8a NAME OF FUNDING/SPONSORING ORGANIZATION Defense Advanced Research Projects Agency		8b OFFICE SYMBOL (if applicable)	10 SOURCE OF FUNDING NUMBERS	
8c ADDRESS (City, State, and ZIP Code) 1400 Wilson Boulevard Arlington, VA 22209-2308		PROGRAM ELEMENT NO 61153N	PROJECT NO RRO4206	TASK NO RR04206-OC
		WORK UNIT ACCESSION NO. 442c022		
11. TITLE (Include Security Classification) Literature Review: Human Benchmarking of Expert Systems				
12. PERSONAL AUTHOR(S) O'Neil, Harold F., Jr.; Ni, Yujing, & Jacoby, Anat				
13a. TYPE OF REPORT Interim	13b TIME COVERED FROM 7/1/89 TO 1/30/90	14. DATE OF REPORT (Year, Month, Day) January 1990	15 PAGE COUNT 48	
16 SUPPLEMENTARY NOTATION				
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD 12	GROUP 05	SUB-GROUP	Artificial intelligence, expert systems, human benchmarking	
19 ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p>This literature review provides a context for using the human benchmarking approach with expert systems. It includes three parts: the current status of expert system evaluation, a search for cognitive instruments, and a parallel comparison of categories between expert systems and cognitive skill instruments. The review suggests that there are different approaches to expert system evaluation including evaluation criteria and evaluation procedures. The literature offers diverse environments for capturing development aspects of expert system evaluation. The review suggests the possibility of developing a psychometric standard for the evaluation of expert systems and documents similarities and differences between cognitive psychology and artificial intelligence, which is important for the human benchmarking approach.</p>				
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Susan Chipman			22b TELEPHONE (Include Area Code) (703) 696-4318	22c OFFICE SYMBOL ONR 1142CS

LITERATURE REVIEW:
HUMAN BENCHMARKING OF EXPERT SYSTEMS

Harold F. O'Neil

Cognitive Science Laboratory
University of Southern California

Yujing Ni

Anat Jacoby

Center for Technology Assessment
UCLA Center for the
Study of Evaluation

January 1990

Artificial Intelligence Measurement System
Contract Number N00014-86-K-0395

Principal Investigator: Eva L. Baker

Center for Technology Assessment
UCLA Center for the Study of Evaluation

ARTIFICIAL INTELLIGENCE MEASUREMENT SYSTEM	
CONTRACT NUMBER N00014-86-K-0395	
DATE: JAN 1990	
BY: [illegible]	
FOR: [illegible]	
[illegible]	
Dist	[illegible]
A1	[illegible]

This research report was supported by contract number N00014-86-K-0395 from the Defense Advanced Research Projects Agency (DARPA), administered by the Office of Naval Research (ONR), to the UCLA Center for the Study of Evaluation. However, the opinions expressed do not necessarily reflect the positions of DARPA or ONR, and no official endorsement by either organization should be inferred. Reproduction in whole or part is permitted for any purpose of the United States Government.

Approved for public release; distribution unlimited

The authors wish to thank other members of the UCLA human benchmarking group: Eva Baker, Robert Brazile, Frances Butler, Kathleen Swigger, and Merlin Wittrock.

BACKGROUND

The UCLA Center for Technology Assessment of the Center for the Study of Evaluation has an existing contract with Defense Advance Research Projects Agency to study human benchmarking methodology for the evaluation of artificial intelligence systems. Systems of interest include vision, natural language, experts system shells, and expert systems. This document provides a literature review which will serve as an intellectual foundation for human benchmarking in the area of expert systems. The literature will be reviewed from two viewpoints: (1) a computer science and software engineering perspective and (2) a cognitive science perspective with a focus on psychological assessment.

The measurement of expert systems development efforts and products challenges existing technologies for software assessment. The challenges stem from the knowledge engineering approach to software development and the nature of the expert system applications typically developed. Standard approaches to verification and validation (V&V) and quality assurance (Q&A) do not adequately address these challenges.

Two kinds of problems result from the novel characteristics of expert system development. The first kind of problem stems from differences in the software engineering process. For example, existing approaches to software evaluation in the military are based on a model of conventional software development. But the development of knowledge-based expert systems may not conform to the phases in the development life cycle as described in DOD-STD-2167A (U.S. Department of Defense, 1987a). For one, expert system development is more distinctly incremental. Specifications are often developed in the course of knowledge engineering. Thus the verification approach of assessing the simple match between preordained specifications and code becomes inappropriate. Even if

a standard verification and validation approach could be taken, the development phases and milestones for expert systems development do not align with those used in conventional software engineering.

The second kind of problem in expert system evaluation is that there is no agreed upon or fully developed methodology for assessing expert system attributes and quality (U.S. Department of Defense, op. cit.). This deficiency can, in part, be attributed to the relative newness of the development approach, as suggested above. But characteristics of the products of these development efforts also pose special problems. Since the output, if not the process, of an expert system is intended to replicate human expert performance, the problem of defining and measuring expertise arises. Performance criteria or standards for expert systems are difficult to define. Hayes-Roth, Waterman and Lenat (1983) note that the assessment of human expertise is just as troublesome, and they cite the need for a "gold standard" or performance criterion against which expert systems can be judged. This need is still present.

A measurement model for expert systems will need to account not only for alternative user contexts, decisions, resources, and goals, but also may serve three functions:

1. Provide approaches to evaluate candidate tools (e.g., "shells"), for those frequent occasions when expert system development is assisted by commercial products.
2. Provide approaches to evaluate the process of development at each stage.
3. Provide approaches to evaluate the quality of the product and process of expert systems.

Any measurement model needs to account for a full range of factors that influence the utility of the resulting information. Among the most salient is the type(s) of decisions of interest. At issue is whether the intent of the evaluation is summative (that is, to make a choice

among competing products, as in steps 1 and 3) or formative (that is, to influence the quality of the process and product as it is developed). A measurement model should also encompass both quantitative and qualitative data collection and analysis techniques. A measurement model should also provide options that are low-cost as well as those that are resource-intensive. The best information can always be obtained at a high cost. What is critical is that a model optimize cost of the measurement process with risk and criticality of the application under development, e.g., exploratory systems may only need low-cost techniques.

When we focus on the quality of the software as it undergoes development and in its finished form (see Table 1), we are forced to consider another set of dimensions that a robust measurement model for expert systems must include. The first set of issues involves looking systematically at the domain of expertise under development. Of concern is the quality of the domain knowledge from both reliability and validity perspectives. On the one hand, one needs procedures to assess the extent to which the knowledge acquired by the system reliably and comprehensively represents the expert(s) participating in the study. On the other, one may be interested in the extent to which the system would solve problems comparably to other experts. In order to accomplish this, one could consider the development of benchmark problem typologies, where specific problems will vary as a function of the domain and each type will incorporate some reasonable combination of critical domain knowledge and reasoning. This approach will require careful feasibility analysis to determine whether such typologies can overcome the varied purposes and domains of expert systems.

Table 1: Quality Concerns for Expert Systems

Acquisition Concern	User Concern	Quality Factor
PERFORMANCE— HOW WELL DOES IT FUNCTION?	How Well Does It Utilize A Resource?	Efficiency
	How Secure Is It?	Integrity
	What Confidence Can Be Placed In What It Does?	Reliability
	How Well Will It Perform Under Adverse Condition?	Survivability
	How Easy Is It To Use?	Usability
DESIGN— HOW VALID IS THE DESIGN?	How Well Does It Conform To The Requirements?	Correctness
	How Easy Is It To Repair?	Maintainability
	How Easy Is It To Verify Its Performance?	Verifiability
ADAPTATION— HOW ADAPTABLE IS IT?	How Easy Is It To Expand Or Upgrade Its Capability Or Performance?	Expandability
	How Easy Is It To Change?	Flexibility
	How Easy Is It To Interface With Another System?	Interoperability
	How Easy Is It To Transport?	Portability
	How Easy Is It To Convert For Use In Another Application?	Reusability

Reprinted from Bowen, Wigle and Tsai (1984)

A measurement model will also make explicit the assessment of unintended outcomes of the expert system development or product. Unintended outcomes may be positive and result in the identification and ratification of new objectives or applications of an implementation, or they may be negative and create negative side effects that dog the application until its replacement. An example of a negative, unintended effect of implementing an expert system is described below.

One potential negative side effect of relying on automation is the danger that apprentice users will not gain expertise of their own as they increasingly rely on such systems. This could result in changes in the availability and, hence, cost of expertise. During the development of expert systems, expertise, which is already rare, must be allocated to the task of knowledge engineering. The increased scarcity of experts can be expected to make them even more costly in the short run. When the system is developed and implemented, however, novice or apprentice users can perform tasks that would otherwise have required the expert. Suddenly expertise is plentiful and cheap. But as users rely on the system, they may fail to gain the skills that would develop them into experts. This could lead to an eventual knowledge dearth in those domains where expertise was originally in short supply. Such a shortage could be even more severe than that which motivated the original development. A fully developed approach to evaluating expert systems must consider indicators of unintended consequences in order for evaluation to assist in the management of expert systems development and implementation.

UCLA/USC personnel have been engaged in developing a characterization of the expert system development process for the past two years. This ongoing project has resulted in a detailed characterization of the development process in terms of stages of development, evaluation considerations, and knowledge engineer question types throughout the process. Documentation has been compiled in a case study methodology on

some differences in the development process with changes in project organization, size, and purpose, application type, knowledge programming environment, and personnel attributes. These differences have implications for the complexity of a development model as well as the kind, purpose, and timing of measurements that might be made during the development.

A brief comparison of the stages of development for one expert system (a psychometric selection consultant) and a conventional software development cycle illustrates some of the differences (see Table 2). We have observed several variations on the expert system development process outlined in the table. The mapping of stages between the two approaches is thus tentative.

One way in which the process differences may lead to differences in measurement methodologies is that expert system development does not lend itself well to the use of written formal specifications. Iterative design and coding tends to be more of a data-driven—as opposed to requirements-driven—enterprise. Many details of functional specification only emerge in the course of development. While many software attributes such as speed requirements or user interface design may be specified in advance, the contents of a knowledge base and the implicit inferencing logic—i.e. the program itself—is the only complete statement of the software's problem-solving capabilities. This absence of specifications means that assessment cannot simply be comparisons of specifications and outputs, or specifications and code. Rule explanations, whether explicitly elicited and coded, or generated by the shell (e.g. via inference trees), constitute the documentation for the origin and purpose of rules. The effect of these rules under different inputs and inference methods determines the problem solving behavior of the system. Thus the task of evaluation of problem solving functionality becomes one of "reverse-engineering" the performance specifications from the functioning knowledge base.

**Table 2. Differences in Development Cycle
Between Experts Systems and Standard Software**

Conventional Software Development	Expert system development
Software requirements analysis	Select domain expert, elicit overview of domain, terminology, etc., select task for application
Preliminary design	Elicit cases, identify functional blocks in consultation process
Detailed design	Map cases onto structure, refine and limit scope (begin coding)
Coding and unit testing	Elicit additional rules and cases, test, refine
CSC integration and testing	Expert review User or field test
CSCI-level testing	Formal testing and certification of expertise

(from Slawson, 1987, and Bowen, Wigle & Tsai, 1984)

A study of the knowledge engineering process by UCLA personnel (Slawson, Hambleton & Novak, 1988) revealed that much knowledge is elicited during the course of development that never ends up in the system. Some of this is deliberate, due to scoping or resource constraints, or rational decisions of the knowledge engineer or programmer. However, some information seems to simply get lost in the process. A metric which might be explored is "How much of the data elicited in the knowledge engineering sessions ended up in the system?" The companion question, "Was this enough?" can be answered in terms

of performance measures. But the first question can also be answered by examination and cataloging of the knowledge base. Where omitted knowledge is important, a standard of performance could be derived from the amount of knowledge a novice learner would have obtained expending similar resources to the knowledge engineering task.

HUMAN BENCHMARKING

Our human benchmarking approach is to establish an evaluation, that is, to norm an expert system's performance on a sample of people's performance. The implication of this approach is that it goes beyond the conventional approach of expert system evaluation and aims to build psychometric criteria through comparison of an expert system's performance with differentiated performances by a sample of people.

To provide a context for our human benchmarking approach, the literature review activities included four themes: the current status of expert system evaluation, a search for cognitive skill instruments, parallel comparison of categories between expert system and cognitive skill instruments, and analysis of the target expert system (i.e., "Gates").

Evaluation of Expert Systems

To investigate the current state of the art in evaluation of expert system, we located a large number of studies that either dealt with expert system evaluation or included evaluation as one component of expert system development. We began the review process by computer searching four data bases, i.e., Education Resource Information Center (ERIC), National Technical Information Service (NTIS), Applied Science & Technology (AST), and University Microfilm Abstracts (UMI) of dissertation abstracts. The searches yielded a set of 103 relevant studies including empirical research, reviews of literature, and

theoretical papers. The keywords used for the search and the results are presented in Table 3. The abstracts themselves of relevant hits are found in Appendix 1.

Table 1. Keywords and Results for the Computer Database Search

System	Dates of Search	Keywords	Hits	Relevant Hits
ERIC	Jan. 1983- Mar. 1989	Expert system & Evaluation Measurement Assessment Comparative study Test Intelligent tutoring system Intelligent computer aided instruction	145	50
NTIS	Jan. 1985- Dec. 1989	Expert system & Evaluation Testing Assessment Measurement Test and evaluation Benchmarking Benchmark Comparative Intelligent tutoring system Intelligent computer aided instruction Knowledge-based tutoring system	151	28
UMI	1985-1988	Expert system	336	23
AST	Oct. 1983- Dec. 1989	Expert system evaluation	3	2

Our analytic review of these reports of evaluation of expert systems can be classified into two major configurations: evaluation criteria and evaluation methods. Evaluation criteria concern what characteristics or attributes of an expert system an evaluation is to capture. Whereas evaluation methods are related to specific procedures of which three primary decisions need to be made, that is, what to validate—process, product, or both, what to validate against—preset criteria or expert performance, and what to validate with—the choice of test cases.

Evaluation Criteria

Evaluation criteria for expert systems usually reflect the goal of the evaluation as being concerned with both the system and its users. Accordingly, evaluation criteria for expert systems falls into a general framework for evaluation of a software system which includes three classical aspects—reliability, validity, and usability. For example, Hollnagel (1989) proposes seven criteria along these three general evaluation principles (see Table 4). These criteria are correctness of the reasoning techniques, sensitivity, robustness, correctness of the final decision, accuracy of the final decision, quality of the human-computer interaction, and cost-effectiveness.

Kaisler (1987) suggests two classes of metrics for evaluation of expert system functions which are summarized in Table 3. One class of metrics is proposed to assess performance during development in terms of problem solving ability. The second one focuses on qualitative evaluations based on quantitative attributes of system. Such metrics can be used to assess size, speed, usefulness, and other criteria.

Kaisler's suggested metrics are adaptations of some conventional software systems metrics to the special qualities of knowledge based systems. He proposes several

categories for metrics which are summarized in Table 5 as well as potential metrics which are summarized in Table 6.

Table 4. Components of System Evaluation
(Adapted from Hollnagel, 1989)

Method Evaluation	Evaluation Criteria
Reliability Technique	Correctness of Reasoning or the internal consistency of the reasoning technique
	Sensitivity or the minimum variation in input needed to change the outcome of the decision
	Robustness or the ability to absorb and compensate for non-standard input
Validity	Correctness of the Final Decision or Output consistent with the needs in the given contexts
	Accuracy of the Final Decision or the extent to which the consequences of alternatives are satisfactory
Usability	The Quality of the Interface or the degree to which the interaction between users and system functions effectively
	The Cost-Effectiveness or the gain from the use of the system related to its cost

Table 5. Attributes for Expert System Evaluation
(adapted from Kaisler, 1987)

Attributes	Issues for Metrics
Evolution and growth	Rapid prototyping continual continual growth in code
Hardware e.g. conventional vs. Lisp machines	Factoring of (complex) rules into several rules affects speed,volume, and size Price/performance Effect of engine qualities on interpretation of execution metrics
Functional improvement	Improved accuracy vs. handing larger, more complex problems Domain specificity of problem complexity metrics Metrics specific to application types (e.g., diagnosis, consultation, planning)
Ease of system redesign	Assessing impact of qualitative changes (e.g., splitting a class in declarative knowledge results in re-assignment of instances)
Trend analysis	Interpretation of incremental snapshots (e.g., more rules may mean more power but also less speed)
Applications vs. technology	Shell vs. application metrics: interrelated performance effects

**Table 6. Potential Matrices for Evaluation of Experts Systems:
A Computer Science Perspective**

Knowledge base size	<p>Represents magnitude of development effort. May include components for declarative and procedural knowledge.</p> <p>Examples: object volume (e.g. number of frames), relative object volume (distribution of classes / instances), average number of slots per class, number of slots having default values vs derived values averaged over the number of instances in a class, weighted average size of instance in terms of number of slots it possesses rule volume, average number of hypotheses and consequents, maximum for any one rule, histogram of rules according to number of hypotheses and consequences.</p>
Knowledge base execution metrics	<p>Efficiency of data management routines to deliver information to execution routines.</p> <p>Examples: slot access time, slot storage time.</p>
Execution metrics	<p>Speed of evaluating procedural knowledge. Indicate quality of programming within shell and amount of knowledge.</p> <p>Examples: number of rules considered per cycle, number of rules selected per iteration, number of rules executed per iteration, number of cycles to solve a problem. Stationary metrics, e.g. number of rules to reach equilibrium after an item of data, time to reach equilibrium</p>
Application granularity	<p>Amount of information in individual rules as well as number of distinct concepts in procedural knowledge.</p> <p>Examples: results of rule base normalization, application vocabulary (e.g. number of operators for rules, number of predicates which may be used in rules, etc.)</p>

(from Kaisler, 1987)

Rothenberg et al.'s work (1987a,b) on the evaluation of expert system shells is closely related to the evaluation of its applications, since shell evaluation must take into account its application characteristics. their preliminarily classified attributes of problems, problem domains, and projects (Table 7). These attributes identify dimensions by which applications attributes might be classified.

Table 7. Characteristics for Expert System Shell Evaluation
(adapted form Rothenberg, 1987)

Problem characteristics
Problem domain
kinds of knowledge
constraints
Problem to be solved within the domain
special processing/knowledge/representation
problem type
other problem attributes
Knowledge acquisition/expertise
characteristics and constraints
Target environment
constraints
end-users
Project characteristics
Scope
goals and budget
Development environment
constraints
Development team
characteristics

Although metrics may not be appropriate for each of these dimensions on any given product a measurement should consider how the overall project and problem characteristics may affect the metrics that are eventually selected. An effort should also be made to keep the metrics, or at least the categories of metrics, relatively independent. Metrics with low intercorrelations would have the advantage of providing more information about the object of analysis providing that each of measures were relevant to the purposes of evaluation. The existence of a logical set of attribute classes, such as that proposed by Rothenberg et al. (1987a,b), should make it easier to keep metrics independent at least on logical, if not on empirical, grounds.

Evaluation Method

Hollnagel (1989) summarizes five categories of evaluation method and their respective advantages compared by referring the criteria proposed previously (Fig.1). These methods focus on the evaluation of an expert system after its implementation.

Statistical sampling. Statistical sampling of test cases. The test cases for the assessment of performance by an expert system are statistically sampled so that the chosen cases are known to be representative (Yu, et al., 1984; Hudson, & Cohen, 1984)

Summative and formative evaluation. Summative evaluation (Scriven, 1967) focuses on overall choices among systems or programs based upon performance levels, time, and cost. This evaluation is essentially comparative and contrasts the innovation against other options. The typical questions the evaluation ask are "Does the intervention work?", "How much it costs?", and "Should we buy it?". Formative evaluation (Baker, 1974) seeks to provide information that focuses on the improvement of the innovation and is designed to assist the developer. Formative evaluation also addresses, from a metaevaluation perspective, the effectiveness of the development procedures used in order to predict whether the application of similar approaches will likely have effective and efficient results (O'Neil, & Baker, 1987). Thus, the formative evaluation seeks to improve the technology at large, rather than the specific instances addressed one at a time.

Analytic hierarchy process. The expert system is decomposed into constituent components to see each part's performance (Liebowitz, 1985).

Whereas Hollnagel's methods focus on the evaluation of an expert system after its implementation, O'Keefe et al.'s (1987) methods of validation of expert system capture the whole development process for expert systems (O'Keefe, et al., 1987) (see Table 8). They categorized validation methods into the qualitative and the quantitative. These validation methods are described in Table 8. The face validation and the predictive validation method

Table 8. Methods for Validation Evaluation

(Adapted from O'Keefe et al. 1987)

Method	Description
Face Validation	Preliminary comparison of system performance with expert performance against a prescribed performance range (???)
Predictive Validation	Assessment of performance by using historic cases and either (1) known results or (2) measures of human expert performance on these cases (???)
Turing Tests	Expert judges' blind evaluation of both system and human expert performance for given cases
Field Tests	Evaluation of prototypical expert system in the intended context
Subsystem Validation	Decomposition of subsystems in an expert system and evaluation of the performance of each subsystem under given input data
Sensitivity Analysis	Validation of system by systematically changing expert system input variable and parameters over some range of interest and observing the effect on system performance
Visual Interaction	An validation environment in which experts' direct interaction with expert system for face validation, subsystem validation, and sensitivity validation
Quantitative validation	Statistical techniques to compare expert system performance against either test cases or human experts

are preliminary approaches to validation during the development of the system. Turing tests and field tests are validation methods used after the installation of the system.

The essential task of expert system evaluation is to translate selected criteria into testable requirements, determine variables or parameters constrained by the requirements, and design test cases for the manipulation of the variables or parameters. However, the most serious problem in carrying out these procedures is that testable requirements are hard to define because relatively few requirements are initially formulated during or after the development of an expert system. Related to this problem is that the profile of representative test cases is usually unknown, thus there is great uncertainty in whether a set of selected test cases represents a reasonable range of example collection.

It is also common to generate test cases to be evaluated by the domain expert during system development. The establishment of a criterion-referenced testing technology for human performance assessment is a relatively recent phenomenon that should not be ignored. Test case generation today is often of the ad-hoc variety, where the expert arbitrarily generates "typical" situations and solutions and the knowledge engineer introduces variations in some parameters in order to test boundary conditions and other variations. The failure to circumscribe the domain in terms of performance criteria makes it difficult to ever know if all of the important variations have been tested. Criterion-referenced testing and item generation technologies used in human performance assessment could be of enormous benefit in designing test case domain specifications that align with applications requirements, specifying input and output conditions, and generating items which effectively sample the domains. Without some systematic approach to test case development, it is difficult to know the limits of an application or to be assured that gaping holes do not lurk within the code.

In an attempt to improve test case generation technique Hall and Heinze (1989) developed a simulation technique. Their test case simulation model (Fig.2) proposes three dimensions—characteristic of model parameter, model parameter, and test case category—for defining any test case.

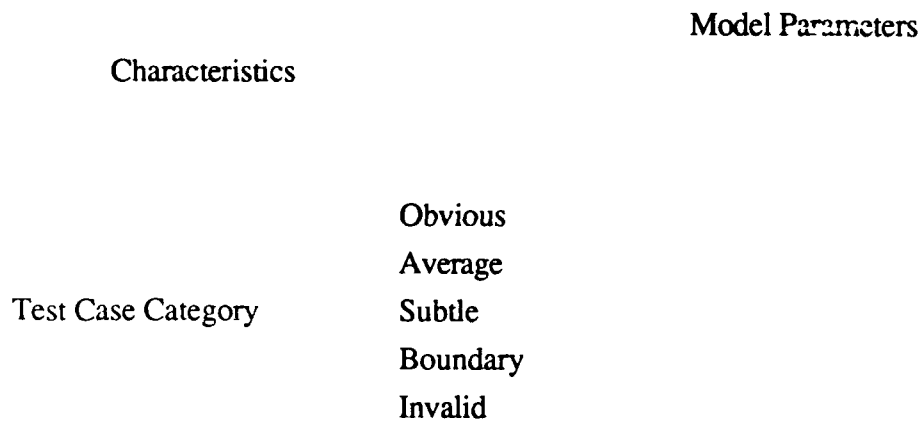


Fig. 2. The Test Case simulation Model Space
(Hall, & Heinze, 1989)

The range of test case collection contains five categories from the obvious to the invalid proposed by Geissman and Schultz (1988). Model parameters indicate identified conditions under which the expert system is to be tested. Characteristic specifies the region of a parameter. For instance, they used this model to generate test cases to assess a Signal Analysis Expert System. An obvious test case is defined as consisting of a direct current signal with a low power level, low noise, no signal dropout, and no interfering signal (Fig. 3).

They suggest that having specified the test cases in this manner, the various categories can be weighted to reflect the degree to which the expert system performance in each category affects its overall performance.

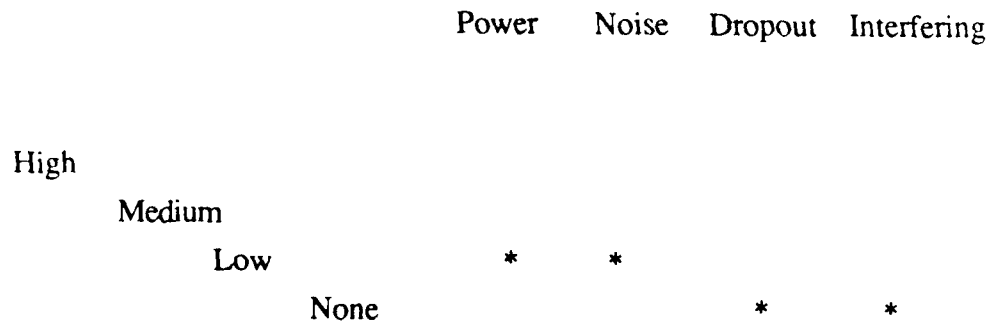


Fig. 3. Definition for an Obvious Test Case.
(adapted from Hall, & Heinze, 1989)

The simulation model for test case generation for expert system evaluation appears to be appealing. It provides an approach to the search for "golden assets"--test case collection--against which to test expert systems. The problem, however, still exists how to define each category of test case, that is, what mean "obvious, average, subtle, boundary, invalid." Because having the definitions explicitly determined depends on a complete specifications for an expert system which has been considered very difficult.

Empirical Studies in Evaluation of Expert System

In the selected set of abstracts (see Table 9) 29 are empirical studies. The methods they employed included Turing test, expert assessment of face validation, criteria assessment with either known results or established test cases, field test, subsystem

validation, human benchmarking, summative and formative evaluation, impact analysis (effects of expert system uses on people's performance and organizations' effectiveness and efficiency such as workload, accuracy, and confidence), and in expert system tool comparative study. These studies are presented in Table 9 in terms of system type, evaluation method, and, if provided, number of test cases.

It is not surprising that most of the studies evaluated systems in the way "do what Charlies does" (Geissman, & Schultz, 1988, pp.28.) since validity of systems was the main concern in these evaluations.

The test cases in these studies either were generated by an expert or experts in correspondence with his or their perceived pre-specification for a system (e.g., Hushon, 1988), or previously used examples in evaluation of similar systems (e.g., Weiss, & Kulilowski, 1988; Tam, et al., 1988). The representativeness of the cases seems to be problematic because the test-case generation technique relies only on the experts' arbitrary judgements which vary across situations and over time. In fact, the representativeness issue occurs during the process of expert system development. The domain knowledge coded in an expert system is usually extracted from one or two experts in the field (e.g. the Gates system's domain knowledge is from an expert). However, there is great variation in domain experts for a given problem, even the same expert may change his solution methods for the same problem over time. Thus, merely using pre-specifications as evaluation criteria and a very small sample of test case make the representative of the test cases uncertain

**Table. 9 Empirical Studies in Evaluation of
Expert System and Expert System Shells**

Expert System Type	Method	No. Cases	Reference
Consultant System for the Disabled	Turing Test	10	Parry, 1986
PREDICT for Forest Pest Management			Schmoldt, 1987
System for Electronic Circuit	Face Validation	5	Baldwin, 1985
EASY for Physical Work Stress Analysis			Chen, 1987
Diagnostic system for the Reading Disabled			Colbourn, 1982
System for Chemical Emergency			Hushon, 1988
Decision Making System For Training Managers			Kearsley, 1988
System for Chemical Analysis	Criteria Assessment	5	Olivero, 1987
ITS for Engineering Mechanics			Richardson, 1986
K-BAS System for Teacher Performance Evaluation			Stevenson, 1987
System for Stowage Planning			Tam, et al., 1988
Automatic Math Modeling			Wang, et al., 1988
Rule-Based Classification System	Field Test	27	Weiss, et al., 1988
MPS System for Inspecting Materials Processing in Space			Anderson, et al.,
Financial Consultant			Phillips, 1987

Table 9 (cont'd)

Expert System Type	Method	No. of Cases	Reference
GRASPERS For Graphic Presentation	Field Test		Tang, 1986
Chess Expert System	Subsystem Validation		Schaeffer, 1986
IRUS Natural Language System	Human Benchmarking		Baker, et al., 1988
MANDATE Consultant System for Education Program	Summative Analysis		Hofmeister, 1988
HAWA MACH-III ITS for Troubleshooting Skills	Summative & Formative		Massey, et al., 1986
Decision Aid System For Auditor Training	Impact analysis		Eining, 1987
Tutoring System			Grabinger, 1988
Decision Support System			Isett, 1987
Expert System Interface			Lamberti, 1987
Consultant System			Perdu, 1988
TPIXIE and MBR Tutoring System for Algebra			Sleeman, et al. 1988
ITS for Basic Algebra			Stasz, 1988
financial Consultant System			Sviokla, 1986
Tubro Prolog	Tool evaluation		Loftin, 1987
Expert System Shells	Example-Based Rule-Based Comparison		Shaw, 1988

Cognitive Skill Instruments

The searching for cognitive skill instruments was intended to assist us in developing a "human benchmarking" approach to the evaluation of expert system. The work began with several cognitive and metacognitive skill, that is, monitoring, problem solving, reasoning, inference, planning, diagnosis, and scheduling. The selection reflected the combined perspectives from cognitive psychology and expert system applications. For example, "problem solving," "reasoning," "inference," and "monitoring" were considered because they are almost the same labels in expert system applications and cognitive psychology, whereas "planning," "diagnosis," and "scheduling" are from categories of expert system applications. Later, we concentrated effort in the areas of reasoning and metacognition. The main source searched for these cognitive skill instruments were the Buros' mental measurements yearbooks (1985-1989) and the Educational Testing Service (ETS) test catalogues (1986-1989). In addition, the Buros's mental measurement institute was contacted recently for information about unpublished tests. In the field, active researchers (e.g., Flavell, Mayer, and) were contacted by Dr. Wittrock for their suggestions and advice. Finally, some relevant studies were located from the ERIC system.

The initial review work revealed the following findings.

Monitoring. There are two kinds of monitoring in cognitive activity. Monitoring, as a specific performance, refers to the cognitive activity of examining displayed status information, both formal (control panel displays) and informal (sounds, vibration, smells, etc.) (Moray, 1986; Parasuraman, 1986). As a general executive process, monitoring is directed at the acquisition of information about and the regulation of one's own ongoing problem solving processes, which is described as "cognitive monitoring" by Flavell (1981), "metacognition" by Sternberg (1985), and "executive decision" by Kluwe (1987).

These two levels of monitoring differ in that the former monitors the external world (e.g., a display) while the latter monitors the internal world (i.e., one's thoughts). Monitoring as an executive process can use information from monitoring one's thoughts of the external world (e.g., a display). In this report, the term "monitoring" refers to the executive process.

One of the possible effects of monitoring function is the ease with which cognitive strategies are transferred to new task demands (Kluwe, 1987). However, frequent monitoring does not necessarily lead to successful performance (Kluwe, 1987). For example, low task proficiency and low confidence may increase frequency of monitoring. Effectiveness of monitoring may be influenced by task proficiency (Hickman, 1977) or expertise. Hickman interviewed two widely read professional persons, asking them to reflect upon their own comprehension processes while reading. Their comments demonstrated a clear sense of purpose for reading, a very active use of identifiable strategies, and an emphasis on relating prior experience and knowledge to material read.

Personal cognitive style may affect monitoring function (Walczyk, & Hall, 1989). They used the Standard Matching Familiar Figures Test (Kagen, et al., 1964) to identify reflective and impulsive subjects. The subjects read passages, each containing incongruent information. The subjects' performance at detecting these contradiction was recorded as index of comprehension monitoring. They found that the reflective children detected significantly more inconsistencies than the impulsive children across grade levels (3-5 grade).

Unfortunately, we have not found a commercially available instrument for measurement of monitoring or metacognition. In some empirical studies monitoring has been usually measured either through the subjects' self-report of their cognitive processes

during their performance or through some questionnaire about strategies they used in their learning and problem solving. We will describe these approaches later.

Reasoning. Reasoning usually includes inference. Because to make an inference is to pass from something that we know to something else that seems to follow from it (Gellatly, 1989). Problem solving is to set up a goal and to carry out a set of operations to reach the goal, which, of course, heavily involves reasoning and inference (Sinnot, 1989). Reasoning has been studied and measured in typical reasoning tasks such as syllogistic (Johnson-Laird, 1983; Rips, 1983) and conditional reasoning (Wason, 1983). and problem solving (Sinnot, 1989). Conditional reasoning refers to that given four conditions $P, \neg P, Q, \neg Q$, the conditional rule "if P then Q " is determined with certainty only under the condition in which it is falsified. A categorical syllogism consists of two premises and a conclusion, each of which describes the relationship between two sets of things. The first premise relates one term, A , to a second term, B ; the second premise relates B to a third term, C ; and conclusion states a relationship, if one exists, between A and C . Thus, reasoning, inference, and problem solving, which were selected as separate categories, could be treated under one reasoning category. Also, there are relatively rich sources of studies dealing with reasoning (see Galotti's review, 1989) and rich source of reasoning tests we will describe later.

Planning. Planning, in cognitive psychology, is viewed as a problem solving approach which is defined as the predetermination of a course of action aimed at achieving a goal (Hayes-Roth, 1988;); it is also considered as one component of metacognition regulating ongoing thinking processes (Beyer, 1988). We have no commercially available test found for planning skills.

Diagnosis and Scheduling. As mentioned above diagnosis and scheduling are from the categories of expert system applications. There are no such categories in cognitive skills and thus no measures.

Instruments for Reasoning

The following reasoning tests may be useful for our project.

Arlin test of formal reasoning. (Arlin, 1984). The specification of this test is based on Piaget's intelligent model (Piaget, & Inhelder, 1967). The test consists of 13 problem situations involving eight formal concepts which are described in Table 10.

Table 10. Components of Arlin Test of Formal Reasoning
(adapted from Arlin, 1984)

Component	Description
Multiplicative Compensations	Understanding gain or loss in one dimension are made up by gains or losses in the other dimensions, for example, width and length are compensatory for a given area
Probability	The ability to develop a relationship between the the confirming and the possible cases, which predicts the probability of an event with a set of data
Correlations	The ability to know if there is or is not a causal relationship and to explain the minority case by inference of chance variables
Combinational Reasoning	The ability to generate all possible combinations of given number of variables, events, or scenarios
Proportional Reasoning	The ability to discover the equality of two ratios which form a proportion.
Forms of Conservation	The ability to deduce and verify certain conservations by observing their effects and inferring their existence
Mechanical Equilibrium	The ability to simultaneously make the distinction between the coordination of two complementary forms of reversibility--reciprocity and inversion
The Coordination of Two or more Systems or Frames of Reference	The relativity concept asking the ability to coordinate two or more systems of reference

Kit of factor-referenced cognitive tests. (Ekstrom, French, & Harman, 1976). This kit is a set of instruments for identifying certain aptitude factors in an factor-analytic model of intelligence. It includes a set of 72 marker tests for 23 cognitive factors.

Among the tests there are three dealing with reasoning skill. They are inductive reasoning, general reasoning, and logical reasoning. These are described in Table 11.

Table 11. Description of Reasoning Factor
(adapted from Ekstrom, et al., 1976)

Factor	Description
Induction	The reasoning ability to form and try out hypotheses that will fit a set of data
General Reasoning	The ability to select and organize relevant information for the solution of a problem
Logical Reasoning	The ability to reason from premise to conclusion, or to evaluate the correctness of a conclusion

Cornell critical thinking test. (Ennis, Millman, & Tomko, 1985). This test contains subtests of induction, deduction, assumption, evaluation of arguments, and interpretation (see Table 12). These are inferred from the definition "critical thinking is the process of reasonably deciding what to believe and do." (reference) The test has two forms. Level X is usable in grades 5-14; (???); level Z is used for gifted or advanced secondary students, college students and other adults. The items in the test are organized in terms of themes; for instance, several items are organized around some conclusions derived from a set of data.

Table 12. Aspects of Critical Thinking In The Cornell Test
(adapted form Ennis, et al., 1985)

Aspects of Critical Thinking	Description
Induction	Determine whether a fact \supports an idea (???)
Deduction	Decide what follows from the reasons given
Credibility	Make a judgment whether a given statement is believable according to given information
Assumptions	Infer what certain ideas taken for granted underlying a statement
Meaning	Figure out the reason why a generated statement is valid(???)

Measurements for Metacognition

Although there are no commercial available instruments for the measurement of metacognition, researchers in this field designed some for their own research purposes. The techniques for measurement of metacognition in empirical studies may be categorized into six kinds. The following are examples of these methods.

Error detection paradigm. This method is mostly used in the measure of reading comprehension monitoring. A short passage contains a single contradiction. The contradictory information is usually not in contiguous sentences. For example, one passage describes cave-dwelling bats that are deaf, toward the end of the passage it is stated that the bats use echoes to locate objects (e.g., Walczyk, & Hall, 1989). The subject is asked to detect the contradictions. The number of correct detection is used as an index of reading comprehension monitoring.

Self-rating scale. The reading awareness interview was designed to assess children's awareness about reading in three areas: evaluating task difficulty and one's own abilities, planning to reach a goal, and monitoring process towards the goal. The interview contained scale items (Jacobs, Paris, 1988; Cross, & Paris, 1988). For example, one monitoring item is "Why do you go back and read things over again?" with three scored choices: a) because it is good practice (1 score); b) because you didn't understand it (2 score); c) because you forget some words (0 score). Jacobs and Paris (1987) suggested that the instrument is sensitive to developmental and instructional differences in children's metacognition about reading.

Questionnaire Inventory. Learning strategy inventories use this form (ref). The question examples are "Do you make a plan when you write essay?", or "How do you manage to remember difficult materials?" (???) (Zimmerman, & Martinez, 1986; 1988).

Thinking-Aloud protocol analysis. Subjects were asked to solve LOGO programming problem. The statements from their thinking-aloud during problem solving were categorized in the scheme of Sternberg's componential intelligence model including these components: deciding the nature of the problem, selecting performance components, combining performance components, selecting a mental representation, allocating resources, monitoring solution, and being sensitive to external feedback (Clements, 1987).

Evaluating relative efficacy of strategies of strategies used. Two memory strategies were introduced to the subjects when they were asked to remember a list of vocabulary. Then, they were asked to evaluate the relative efficacy of the strategies according to their memory experience (Brigham, & Pressley, 1988).

Behavior observation. Subjects were requested to solve puzzles under reversible and irreversible conditions (the irreversible condition referred to that once a piece

of the puzzle was placed on the working cardboard it became fixed and could not be removed). Changing the problem solving condition would cause children to increase the intensity and to decrease the speed of their solution approach. The change of actions was viewed as the evidence of monitoring and regulating the course of their problem solving processes (Kluwe, 1987).

Parallel Analysis of Cognitive Skill Instruments and Expert System Categories

To use a cognitive skill instrument to norm "intelligence" of an expert system one primary task was to examine the possible overlappings between categories of expert systems and types of cognitive skill instruments. Only when a particular category or function of expert systems is identified to be parallel or not parallel to a particular type of cognitive skill instrument, can the decision be made that the particular instruments are legitimate to be used for norming the performance of expert systems on that of human beings, and that this human benchmarking approach is valid. Both general and specific approaches were used to accomplish this task. The general approach used two category systems of expert systems (Chandrasakaran, 1986; Shalin, Wisniewski, & Levi, 1988), whereas the specific approach employed an expert system (GATES). Their possible correspondences were examined based on the descriptions or definitions of the categories or functions of expert systems and cognitive skill instruments.

General Approach

For the general approach three category systems were reviewed (Swigger, 1989). Waterman's system (1986) classifies expert systems into ten categories in terms of types of tasks expert systems are designed to perform. They are interpretation, prediction, monitoring, diagnosis, debugging, repair, instruction, design, planning, and control. Shalin et al.'s system (1988) categorizes expert systems according to their functions and

knowledge requirements. These categories are classification, interpretation, design, and problem solving and planning. They are described hierarchically inclusive because the functions and the knowledge requirements for more complex expert system functions subsume requirements for less complex expert systems. Chandrasekaran's system (1986) identifies five critical functions or features called "generic tasks"--hierarchical classification, hypothesis matching or assessment, abductive assembly, hierarchical design by plan selection, and state abstraction--through the analysis of a diagnosis and a design systems. If two dimensions--task and skill--were used to describe the features of these category systems, Waterman's is task-oriented while Chandrasekaran's skill-oriented. Shalin's is in the middle. Because the "generic task" analysis seems to be more able to capture similar functions different expert systems may have, which corresponds to our approach to evaluation methodology of expert system, and because almost all cognitive skill instruments focus on skills rather tasks, Chandrasekaran's and Shalin et al.'s systems were chosen for the analysis of possible relations between their categories and cognitive instruments (Table 13).

**Table 13. Parallel Analysis between Expert System Function
Classification and Cognitive Skill Instruments**

Expert System Function Classification	Cognitive Skill Instrument
Shalin et al. (1988)	
Classification (Matches the input features of an exemplar of a class to a concept)	The Kit Subtests: Deduction Induction
Interpretation (Construct a coherent representation from classified objects)	The Kit Subtests: Deduction Induction Reading Comprehension
Design (Arranges objects according to (constraint on these objects)	Pattern Configuration
Problem-solving and Planning (Arranges actions according to constraints on action sequences)	The Arlin Test The Kit Subtests: Figural flexibility Spatial Scanning
Chandrasekaran (1986)	
Hierarchical Classification (Organize concepts in terms of their relations with the top- most concept having control over the sub-concepts)	The Piagetian Class- Inclusion Test The Kit Subtest: Deduction Induction
Hypothesis Matching or Assessing (Generate a concept, match it against relevant data, and determine a degree of fit)	The Cornell Subtest: Credibility Watson-Glaser Test: Inference
Hierarchical Design by Plan Selection and Refinement (Choose a plan based on some specifi- cation, instantiates and executes parts of the plan, which in turn suggests further details of the design)	The Kit Subjects: Figural flexibility Spatial scanning
State Abstraction (Predict a state change when a proposed action may be executed)	Prediction Tests

We examined the definitions of expert system categories and that of cognitive skills or factors an instrument is intended to measure, then to determine in what ways they are or are not parallel.

The results showed that the parallel relation between them are not simple. For the instance of classification, the induction and the logical reasoning tests from the Kit of Factor-Referenced Test (Ekstrom, et al., 1976) have been chosen. The induction test measures the ability to form and try out hypotheses that will fit a set of data. It asks subjects to find concepts of classes which will group all the given objects into these classes. The nature of the task seems to be in correspondence with Shalin, et al.'s definition for classification as matching the input features of an exemplar of a class to a concept. The logical reasoning test identifies the ability to reason from premise to conclusion. One task of the test requires the constructing of hierarchical relations of classes, which is a very matched test for Chandrasenkaran's hierarchical classification category that is defined as organizing concepts in terms of their relations with the top-most concept having control over the sub-concepts. However, there is somewhat subtle in that a classification activity for human being includes concept formation and concept identification (e.g., Vygotsky, 1978;). Concept formation employs a bottom-up (forward chaining in artificial intelligence terminology) approach, while concept identification uses a top-down (backward chaining) approach. The classification function in expert system application usually only contains top-down approach (e.g., Chandrasekaran, et al.'s (1983) MDX medical diagnostic system; Brazile, & Swigger's (1988) gate assignment system).

In addition to this, the data from developmental psychology (provided by Dr. M. Wittrock) tells that the classification in terms of objects of function is viewed as a higher level than that in terms of the objects' shape. But the same judgment may not be made in expert system applications. It may be inappropriate to say that an system doing shape

classification is "smarter" than that doing function classification because both carry out the same function in a technical sense.

The analysis of the example above suggests both the possibility to use cognitive skill instruments to norm expert system performance and the subtle distinctions between them. Further detailed analysis is needed to get a clearer sense of this.

Analysis of the Gates System

Gates is an expert system for gate assignment at TWA's JFK and St. Louis airport (Brazile, & Swigger, 1988; 1989). It has been chosen as a target system for our human benchmarking approach for two reasons. First, the system has some features (e.g. monitoring functions) we are interested in for the benchmarking evaluation of expert system. Second, we have good cooperative relation with the developers of the system and they also show interest in our project, thus, it is convenient for us to reach all the document as well as the details about the development of this system. In this section we will provide a brief description of the system and a parallel analysis between the system's functions and Shalin et al.'s (1988) and Chandrasekaran's (1986) category.

The Gates System

Gates (Brizile, & Swigger, 1988; 1989) is a constraint satisfaction expert system developed to create the monthly gate assignment. Obtained from an experience ground controller, the domain knowledge is represented in Prolog predicates as well as several rule-like data structures including permission rules (the GATEOK predicate) and denial rules (the conflict predicate). These two kinds of rule determine when a set of gates can or cannot be assigned to a particular flight respectively.

The system uses the following procedures to produce monthly gate assignments

1. Considering an unassigned flight that has the most constraints first (a set of FLIGHT rules);
2. Selecting a particular gate for a particular flight by using a set of GATEOK rules that have been arranged in some priority;
3. Verifying whether the gate assignment is correct by checking it against a set of CONFLICT rules;
4. Making adjustments by relaxing constraints to have all flights assigned gates;
5. After all assignments are made, adjusting assignments to maximize gate utilization, minimize personnel workloads, maximize equipment workload.

The Gates Taxonomy and Expert System Categories

To bridge the above parallel analysis work with the GATES system, that is, how the Gates' components or functions could fit these identified instruments, we asked one developer of the system to make a parallel analysis of the Gates' components and Shalin et al.'s (1988) and Chandrasekaran's (1986) expert system category. This analysis is summarized in Table 14.

**Table 14. A Parallel Analysis of Gates Taxonomy
and Expert System Function Classification**

Function Classification	Gates Function
Shalin et al.	
Classification	Classify input feature of plane type
Interpretation	Infer the schedules given data about plane type, and other descriptions
Design	Configure better schedule using constraints of plane type, arriving and departing times
Problem-solving and Planning	Planning operators--two types of rule
Chandrasekaran's	
Hierarchical Classification	Not fit (??)
Hypothesis Matching or Assessment	Process the three passes by which the system keeps refining its hypothesis and produce a better schedule
Hierarchical Design by Plan Selection and Refinement	Assign first flights and gates with the most constraints, then relax constraints to have more flights assigned
State Abstraction	Not fit (??)

The review discussed demonstrates that (1) there are different approaches including evaluation criteria and evaluation procedures to expert system evaluation. The literature offers diverse environments to capture developmental aspects of expert system evaluation; (2) the parallel analysis suggests the possibility to develop a psychometric standard to

evaluation of expert system; the analysis also helps to recognize relations and distinctions between cognitive psychology and artificial intelligence, which is important for our human benchmarking approach as well as effective communication between the two fields.

Based on the above literature and expert judgement we are going to benchmark the performance of the Gates system using an experimental methodology. Several theoretical and technical advances are needed to be tackled to pursue this goal (Baker, 1989), which will be reported in our next report.

Alternative approaches to evaluating expert systems involve the following: (a) the use of expert performance ratings, (b) use of test cases or benchmarks, and (c) effect on job performance, as was done on the MYCIN project (Shortliffe, 1976). A number of expert raters or end users may make judgements about system performance, preferably in blind designs to avoid bias. When multiple raters are used, validation should include the assessment of interjudge consistency of ratings. For such metrics, generalizability theory, or "g-theory" (Brennan, 1983; Cronbach, et al., 1972), holds great promise. G-theory has been used successfully in military performance testing for analogous purposes (i.e. minimizing sources of variance with multiple raters and performance items) but has never been applied to expert system assessment.

Perhaps the ultimate test of an expert system is its effect on performance of the job it performs or assists its user to accomplish. Assessment of expert systems via job performance can be done using a controlled experimental method in a real or simulated task environment. Quality of decisions made on a diagnostic consultation task, for example, can be measured by expert ratings when the task is performed by an unaided novice, by users with various abilities aided by the software, or by experts alone. Significant differences with and without the software indicate its effectiveness or lack thereof.

Significant interactions between software presence/absence and user characteristics would indicate important factors to consider for training and selection of users when the application is fielded.

Some of the more interesting recent work in evaluation of rule-based systems comes from the cognitive arena. For example, Lehner et al. (1985) note that military expert systems involve ill-specified knowledge bases where human experts differ considerably in their opinions. Solutions usually involve merging expertise of multiple human experts with differing areas of expertise. When military applications are embedded in background systems, users are often not familiar with the specifics of the problem being addressed. Thus Lehner characterizes the user's environment as one where multiple problem solvers are trying to cooperatively solve a common decision problem. The different problem solvers have access to different decision processes, heuristics, or data. Lehner's research suggests the importance of the user having an appropriate mental model of the system's process, especially when differing processes are involved. This kind of evidence suggests that the user must be considered part of the system when evaluating its performance. When expert systems perform in consultation with humans, the combined man and machine may become the unit of analysis in measurement. Since the performance of expert systems depends significantly on the users, human assessment becomes a critical dimension of measurement. This will result in the development of metrics sensitive to the human performance dimensions of expert system measurement.

REFERENCES

- Arlin, P.K. (1982). *Arlin test of formal reasoning*. East Aurora, NY: Slosson Educational Publication.
- Baker, E. L. (1974). Formative evaluation in instruction. In J. Popham (Ed.), *Evaluation education*. Berkeley, CA: McCutchan.
- Baker, E.L. (1989). *Human benchmarking*. Center for Technology Assessment, University of California, Los Angeles.
- Beyer, B.K. (1988). *Developing a thinking skills program*. Boston, MA: Allyn & Bacon, Inc.
- Bowen, T.P., Wigle, G.B., & Tsai, J.T. (1984). *Specification of software quality attributes* (RADC-TR-85-37, Vol. I). Griffiss Air Force Base, NY: Rome Air Development Center.
- Brazile, R., & Swigger, K. (1988). GATES: An expert system for airlines. *IEEE Expert*, 3.
- Brazile R., & Swigger, K. (1989). *Extending the GATES scheduler: Generalizing gate assignment heuristics*. Unpublished manuscript.
- Brennan, R. (1983). *Elements of generalizability theory*. Iowa City: ACT.
- Brigham, M., & Pressley, M. (1988). Cognitive monitoring and strategy choice in younger and older adults. *Psychology and Aging*, 3, 249-257.
- Cronbach, L., Gleser, G., Nanda, H., & Rajaratnam, N. (1972). *The dependability of behavioral measurements: Theory of generalizability for scores and profiles*. New York: Wiley.
- Chandrasekaran, B. (1986). Generic tasks in knowledge-based reasoning. High-level building blocks for expert system design. *IEEE Expert*, 1, 23-30.
- Chandrasekaran, B, Mittal, S., Gomez, F., & Smith, J. (1979). An approach to medical diagnosis based on conceptual structures. *Proc. Six Int'l Joint Conf. Artificial Intelligence*, Aug. 134-142.
- Clements, D.H. (1987). Measurement of metacomponential processing in young children. *Psychology in the School*, 24, 23-30.
- Ekstrom, R.B., French, J.W., & Harman, H.H. (1976). *Manual for kit of factor-referenced cognitive tests*. Princeton, NJ: Educational Testing Service.
- Ennis, R., Millman, J., & Tomko, T. (1985). *Cornell Critical Thinking Tests level X and level Z*. Pacific Grove, CA: Midwest Publication.
- ETS Collection Catalog, Vol. 1, 2, 3 (1986-1989). Phoenix, AZ: Oryx Press.

- Flavell, J. (1981). Cognitive monitoring. In W. Dickson (Ed.), *Children's oral communication*. New York: Academic Press.
- Galotti, K. (1989). Approaches to studying formal and everyday reasoning. *Psychological Bulletin*, 105, 331-351.
- Geissman, J.R., & Schultz, R.D. (1988). Verification and validation of expert systems. *IEEE Expert*, 3, February, 26-33.
- Hall, D.L., & Heinze, D.T. (1989). The use of simulation techniques for expert system test and evaluation. *ISA Transactions*, 28, 19-22.
- Hayes-Roth, B. (1988). A cognitive model of planning. In A. Collins, & E. E. Smith (Eds.). *Readings in cognitive science: a perspective from psychology and artificial intelligence*. Mateo, CA: Morgan Kaufmann Publishers.
- Hickman, J. (1977). What do fluent readers do? *Theory into Practice*, 16, 371-375.
- Hollnagel, E. (1989). Evaluation of expert system. In G. Guide & C. Tasso (Eds.), *Topics in expert systems design: Methodologies and tools..* New York: Elsevier Science Publishers, North-Holland.
- Hudson, D.L., & Cohen, M.E. (1984). *EMERGE, a rule-based clinical decision making aid*. IEEE Computer Society, First Conference on Artificial Intelligence Applications, Denver, CO.
- Hushon, J.M. (1988). *The feasibility of constructing an expert system to assist first responders to chemical emergencies*. University Microfilm Abstracts of dissertation abstracts.
- Jacobs, J., & Paris, S. G. (1987). Children's metacognition about reading: Issues in definition, measurement, and instruction. *Educational Psychologist*, 22, 255-278.
- Johnson-Laird, P.N. (1983). *Mental Model*. Cambridge, MA: Harvard University Press.
- Kagan J., Rosman, B.L., Day, D., A.bert, J., & Phillips, W. (1964). Information processing in the child: Significance of analytic and reflective attitudes. *Psychological Monographs*, 78, (1, Whole No. 578).
- Kluwe, R.H. (1987). Executive decisions and regulation of problem solving. In F.E. Weinert, & R.H. Kluwe (Eds.), *Metacognition, motivation and understanding*. Hollsdale, NJ: Lawrence Erlbaum.
- Lehner, P., Zirk, D., Hall, R., & Adelman, L. (1985). *Human factors in rule-based systems: Final report*. Arlington, VA: Office of Naval Research.
- Liebowitz, J. (1985). *Evaluation of expert systems: An approach and case study*. IEEE Computer Society, Second Conference on Artificial Intelligence Applications, Miami, FL.
- Mental Measurements Yearbook*. (1985-1989). Lincoln, NE: University of Nebraska Press.

- Moray, N. (1986). Monitoring behavior and supervisory control. In K.R. Boff, & J.P. Thomas (Eds.), *Handbook of perception and human performance*. New York: John Wiley.
- O'Keef, R.M., Balci, O., & Smith, E.P. (1987). Validating expert system performance. *IEEE Expert*, Winter, 81-90.
- O'Neil, H.F., Jr., & Baker, E.L. (1987). *Issues in intelligent computer-assisted instruction*. Center for the Study of Evaluation, University of California, Los Angeles.
- Parasuraman, R. (1986). Vigilance, Monitoring and search. In K.R. Boff, & J.P. Thomas (Eds.), *Handbook of perception and human performance*. New York: John Wiley.
- Piaget, J., & Inhelder, B. (1969). *The Psychology of the child*, trans. H. Weaver. London: Routledge & Kegan Paul Ltd.
- Rips, L. (1983). Cognitive processes in propositional reasoning. *Psychological Review*, 90, 38-71.
- Rothenberg, J., Paul, J., Kameny, I., Kipps, J., Swenson, M. (1987a). *Evaluating expert system tools: A framework and methodology*. Santa Monica, CA: RAND Corp.
- Rothenberg, J., Paul, J., Kameny, I., Kipps, J., Swenson, M. (1987b). *Evaluating expert system tools: Workshops*. Santa Monica, CA: RAND Corp.
- Scriven, M. (1967). The methodology of evaluation. In R.W. Tyler, R.M. Gagne, & M. Scriven (Eds.), *Perspectives of curriculum evaluation. AERA Monograph Series on Curriculum Evaluation, No.1*. Chicago: Rand McNally.
- Shalin, V.L., Wisniewski, E.J., & Levi, K.P. (1988). A formal analysis of machine learning systems for knowledge acquisition. *International Journal of Man-Machine Studies*, 29, 429-446.
- Shortliffe, E. H. (1976). *Computer-based medical consultations: MYCIN (Artificial Intelligence Series)*. Elsevier, NY: Elsevier Computer Science Library.
- Sinnott, J.D. (1989). AN overview—if not a taxonomy—of everyday problems used in research. In T.D. Sinnott (Ed.), *Everyday problem solving: Theory and applications*. New York: Praeger.
- Slawson, D. (1987, November). *Knowledge engineers: Expert learners?* Paper presented at the Annual Meeting of the California Educational Research Association, San Jose, CA.
- Slawson, D., Hambleton, R., & Novak, J. (1988, April). *Qualitative valuation of expert system shells*. Paper presented at the Annual Meeting of American Educational Research Association, New Orleans.
- Sternberg, R.J. (1985) Cognitive approaches to intelligence. IN B.B. Wolman (Ed.), *Handbook of intelligence: Theories, measurements and applications*. New York: John Wiley.

- Swigger, K. (1989). *Classification schema for expert systems*. Unpublished manuscript.
- Tam, G.W., Kotras, T.V., & Dillingham, J. (1988). *Prototype expert system containership for stowage planning*. Abstract from National Technical Information Service.
- U.S. Department of Defense. (1987). *Defense system software development*. (And companion documents, DOD-SDD-2167A). Washington, D.C.: Author.
- Vygotsky, L.S. (1978). *Language and thought*. Cambridge, MA: Harvard University Press.
- Wason, P.C. (1983). Realism and rationality in the selection task. In J.St.B.T. Evans (Ed.), *Thinking and reasoning: Psychological approaches*. London: Routledge & Kegan Paul.
- Walczyk, J.J., & Hall, V.C. (1989). Is the failure to monitor comprehension an instance of cognitive impulsivity? *Journal of Educational Psychology*, 81, 294-296.
- Waterman, D. (1986). *A guide to expert system*. Reading, MA: Addison-Wesley.
- Weiss, S.M., & Kulikowski, C.A. (1988). *Empirical analysis and refinement of expert system knowledge bases*. Abstract from National Technical Information Service.
- Yu, V.L., Fagan, L.M., Bennett, A.w., Clancey, W.J., Scott, A.C., Hannigan, J.F., Buchanan, B.G., & Cohen, S.N. (1984). An evaluation of MYCIN's advice. IN B.G. Buchanan & E.H. Shortliffe (EDs.), *Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project*. Reading, MA: Addison-Wesley.